



**POLITECHNIKA POZNAŃSKA**

**INSTYTUT AUTOMATYKI**

60-965 Poznań, pl. M. Skłodowskiej-Curie 9  
tel. 320-21 w. 321, 325

Zleceniodawca <b>Zarząd Dróg i Mostów Poznań, ul. Wilczak 16</b>	Nr umowy <b>IA-U/11/74/75 B</b>
Nazwa opracowania <b>Organizacja logiczna komputera wirtualnego WIK</b>	Rodzaj opracowania <b>Instrukcja</b>
Urządzenie	Egz. <b>4</b>
	Nr archiwalny <b>76005</b>

Witold Wojciechowski, Jerzy Bartoszek, Jacek Martinek

ORGANIZACJA LOGICZNA KOMPUTERA WIRTUALNEGO WIK

Zleceniodawca : Zarząd Dróg i Mostów  
Poznań, ul. Wilczak 16

Znak zlecenia : TRE / 7132 / 75

Nasz znak : IA - U / 11 / 74 / 75 B

Dyrektor Instytutu

  
doc. dr hab. inż. Antoni Woźniak

Poznań 1976

SPIS TREŚCI

1. Wstęp
2. Organizacja logiczna komputera wirtualnego WIK
  - 2.1. Opis ogólny WIK-a
  - 2.2. Organizacja pamięci wirtualnej
  - 2.3. Synchronizacja i przesyłanie informacji pomiędzy procesorami
  - 2.4. Struktura i działanie wirtualnego procesora arytmetyczno-logicznego; lista rozkazów
  - 2.5. Struktura i działanie wirtualnego procesora we/wy
  - 2.6. Struktura i działanie wirtualnego procesora zewnętrznego.

## 1. Wstęp

Wirtualny komputer WIK z punktu widzenia programisty-użytkownika jest maszyną cyfrową, której podstawowymi cechami są:

- wieloprocesowość,
- równoczesność operacji we/wy i obliczeń,
- łatwość programowania zadań sterowania w czasie rzeczywistym.

WIK zrealizowany jest na sprzęcie systemu Momik 8b za pomocą symulacji programowej. Rozkazy WIK-a wykonywane są przez podprogramy zapisane w instrukcjach Momika 8b. Strukturę i semantykę rozkazów WIK-a dobrano w ten sposób, by usunąć niedogodności programowania Momika 8b, którymi są:

- trudny dostęp do pamięci (jeden rejestr strony),
- mała pojemność strony (32 słowa),
- uboga lista rozkazów,
- trudna obsługa transmisji znakowych w kanale arytmometru,
- trudne programowanie obsługi przerw.

Bogatsza lista rozkazów umożliwia programowanie w zwięzły sposób, co powoduje zmniejszenie wymagań odnośnie pojemności pamięci.

Odbywa się to kosztem konieczności rozpoznawania programowego rozkazów WIK-a, co powoduje zmniejszenie szybkości działania.

Rozkazy arytmetyczno-logiczne oraz synchronizacyjne umożliwiają łatwe zakodowanie nawet dość skomplikowanych algorytmów sterowania w czasie rzeczywistym wymagających obliczeń typu arytmetycznego i decyzji typu logicznego oraz wprowadzania danych, wyprowadzania wyników sterujących i raportów informacyjnych.

Aby umożliwić optymalny dobór rozkazów wykorzystywanych przez użytkownika przewidziano możliwość uzupełnienia aktualnej listy rozkazów rozkazami nowymi, a nawet nowymi grupami rozkazów.

Wykorzystanie koncepcji pamięci wirtualnej zapewniło użytkownikowi możliwość korzystania z bazy danych o wielkości dostosowanej do jego potrzeb i niezależnej od wielkości pamięci operacyjnej systemu Momik 8b.

Wykorzystując wirtualny komputer WIK można zaprogramować sterowanie w czasie rzeczywistym obiektem, który spełnia warunki wymagane przez system Momik 8b, a w szczególności przez kanały wprowadzania i wyprowadzania informacji oraz system przerwań.

## 2. Organizacja logiczna komputera wirtualnego WIK

### 2.1. Opis ogólny WIK-a

Zasadniczymi częściami składowymi WIK-a są:

- pamięć wirtualna,
- strony sterowania,
- wirtualne procesory arytmetyczne,
- wirtualne procesory we/wy,
- wirtualne procesory zewnętrzne.

Pamięć wirtualna składa się ze słów 8-bitowych i służy do przechowywania rozkazów oraz danych. Wielkość pamięci wirtualnej uzależniona jest od ilości pamięci dyskowych (max.4) wykorzystanych przy jej realizacji.

W szczególności może być ona zrealizowana w oparciu tylko o pamięć operacyjną.

Strony sterowania wykorzystywane są do przechowywania informacji o procesorach oraz do przesyłania informacji między procesorami i są dla programisty niedostępne.

Wirtualnym procesorem nazywać będziemy abstrakcyjną maszynę, która zdolna jest do wykonywania (interpretowania) rozkazów WIK-a lub prowadzenia transmisji danych. Pierwsze z nich nazywamy wirtualnymi procesorami arytmetyczno-logicznymi, a drugie wirtualnymi procesorami we/wy oraz wirtualnymi procesorami zewnętrznymi. Każdy wirtualny procesor jest identyfikowany poprzez swoją nazwę (identyfikator), która jest liczbą 6-bitową.

Niezależnie od tego, każdy wirtualny procesor arytmetyczno-logiczny posiada priorytet, który jest liczbą z zakresu 0 do 255. Ilość, rodzaje, nazwy oraz priorytety procesorów wirtualnych są ustalane przez programistę.

Procesor wirtualny może znajdować się w jednym z następujących stanów: przetwarzania, zablokowania, zatrzymania oraz zablokowania i zatrzymania.

Związek między wirtualnymi procesorami arytmetyczno-logicznymi a procesorem fizycznym jest następujący: zbiór wirtualnych procesorów arytmetyczno-logicznych jest zrealizowany na procesorze fizycznym metodą podziału czasu procesora fizycznego pomiędzy wirtualne procesory arytmetyczno-logiczne. W ten sposób jeden fizyczny procesor, na którym zrealizowano  $n$ -procesorów wirtualnych widziany jest przez użytkownika jako  $n$  niezależnych procesorów  $n$  razy wolniejszych od fizycznego (o ile czas procesora fizycznego dzielony jest równomiernie).

Interpretacja procesorów we/wy i procesorów zewnętrznych jest następująca: każde urządzenie zewnętrzne (z wyjątkiem urządzeń zewnętrznych dołączonych do kanałów bezpośredniego dostępu) jest związane z jednym procesorem we/wy lub procesorem zewnętrznym. Związek ten jest wzajemnie jednoznaczny. W przypadku kanału bezpośredniego dostępu wszystkie urządzenia zewnętrzne dołączone do niego związane są z jednym procesorem we/wy lub procesorem zewnętrznym.

## 2.2. Organizacja pamięci wirtualnej

Jak wspomniano w punkcie 2.1 pamięć wirtualna składa się ze słów 8-bitowych. W celu uniezależnienia adresacji pamięci wirtualnej od jej wielkości została ona podzielona na segmenty o pojemności 256 słów. W ramach segmentu słowa numerowane są od 0 do 255. Segmenty identyfikowane są przez nazwę (identyfikator) segmentu. Wielkość identyfikatora uzależniona jest od

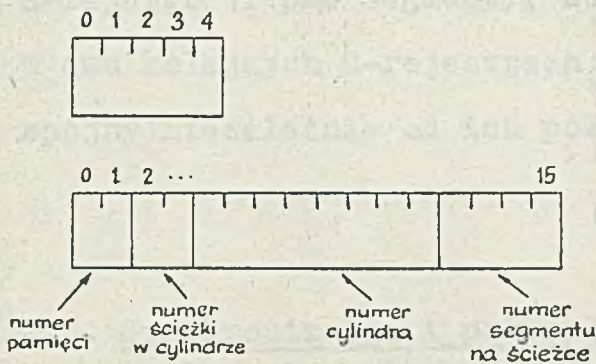
sposobu realizacji pamięci wirtualnej.

Przy realizacji w oparciu tylko o pamięć operacyjną systemu Momik 8b identyfikator jest liczbą 5-bitową (max.32 segmenty). Jeśli wykorzystano przy realizacji pamięci wirtualnej pamięci dyskowe, identyfikator jest liczbą 16-bitową i ma on strukturę związaną z adresacją pamięci dyskowych. W tym przypadku maksymalna ilość wszystkich segmentów jest równa:

$$4 \cdot x \cdot 4 \cdot x \cdot 203 \cdot x \cdot 16 = 51968$$

(pamięci) (ścieżki) (cylindry) (segmenty na ścieżce)

Rysunek 1 przedstawia strukturę identyfikatora 5 oraz 16-bitowego.



Rys.1. Struktura identyfikatora segmentu 5-cio i 16-bitowego.

### Adresacja pamięci wirtualnej

Z każdym wirtualnym procesorem arytmetyczno-logicznym oraz procesorem we/wy związane są cztery 16-bitowe rejestry zwane rejestrami bazowymi lub S-rejestrami. W każdym rejestrze bazowym wirtualnego procesora arytmetyczno-logicznego programista może umieszczać (wykorzystując odpowiedni rozkaz) iden-



tyfikator segmentu. Dla danego S-rejestru operacja ta może być wykonywana wielokrotnie. Umieszczanie identyfikatorów segmentów w rejestrach bazowych wirtualnych procesorów we/wy odbywa się automatycznie.

Rejestr bazowy (danego procesora wirtualnego), w którym nie umieszczono identyfikatora segmentu nie może być wykorzystywany do adresacji.

2-bitowy numer S-rejestru oraz 8-bitowy numer słowa w segmencie składają się na 10-bitowy adres względny. Posługując się 10-bitowym adresem względnym programista może adresować pamięć względną o maksymalnej pojemności 1024 słów. Umieszczenie pamięci względnej w pamięci wirtualnej zależy od zawartości S-rejestrów. Dwa segmenty, których identyfikatory umieszczono w dwu kolejnych S-rejestrach traktowane są zawsze jako obszar spójny niezależnie od ich położenia w pamięci wirtualnej.

### 2.3. Synchronizacja i przesyłanie informacji pomiędzy procesorami

Procesory wirtualne powinny mieć możliwość współdziałania i wzajemnego porozumiewania się. Aby opisać tę współpracę posłużymy się pojęciem zdarzenia.

Zdarzeniem nazywamy zaistnienie w dowolnym procesorze wirtualnym takiej sytuacji, o której powinien zostać poinformowany jakiś inny procesor. Przykładami zdarzeń są: zakończenie transmisji danych przez wirtualny procesor we/wy, wystąpienie błędu w czasie działania procesora wirtualnego itp.

O zdarzeniach procesory wirtualne informują się przez wysyłanie komunikatów i otrzymywanie odpowiedzi. Komunikaty i odpo-

wiedzi, które nadesłano do danego procesora wirtualnego umieszczane są w kolejce komunikatów i odpowiedzi związanej z tym procesorem. Kolejność i sposób wykorzystania informacji w nich zawartej zależy od działania procesora wirtualnego. Do przesyłania komunikatów i odpowiedzi wykorzystywane są strony sterujące. Jedna strona sterująca może być wykorzystana do przesłania jednego komunikatu lub jednej odpowiedzi. Pojemność strony sterującej wynosi 32 słowa. Ponieważ w komputerze WIK ilość stron sterujących jest ograniczona, może się zdarzyć, że wirtualny procesor zgłaszający zapotrzebowanie na wolną stronę sterującą będzie musiał na nią czekać przez pewien okres czasu.

Procesory czekające na wolne strony sterujące zorganizowane są również w postaci kolejki. Kolejka ta obsługiwana jest na zasadzie "pierwszy przyszedł - pierwszy zostanie obsłużony".

W czasie oczekiwania na wolną stronę sterującą procesor wirtualny jest zawsze w stanie zablokowania lub zablokowania i zatrzymania.

Komunikaty i odpowiedzi składają się z dwu zasadniczych elementów: nazwy nadawcy oraz treści. Nazwę nadawcy zawiera zawsze drugie słowo strony sterującej, natomiast słowa od trzeciego do trzydziestego drugiego zawierają treść. Treść komunikatu lub odpowiedzi może składać się zatem maksymalnie z 30 słów. Zawartość pierwszego słowa treści ustalana jest przez komputer wirtualny automatycznie (bardziej szczegółowe informacje na ten temat można znaleźć przy opisach konkretnych sytuacji wysyłania lub otrzymywania komunikatów i odpowiedzi).

W komunikatach i odpowiedziach wysyłanych przez wirtualne procesory arytmetyczno-logiczne o zawartości pozostałych słów decyduje programista.

Koniec komunikatu lub odpowiedzi wskazuje słowo o zawartości równej  $12_8$  (w kodzie ISO-7 jest to znak "nowa linia").  
Niezależnie od synchronizacji przy pomocy komunikatów i odpowiedzi współpraca wirtualnych procesorów arytmetyczno-logicznych jest organizowana przez komputer wirtualny automatycznie. Współpraca ta jest organizowana tak, aby procesor fizyczny był wykorzystywany przez wirtualny procesor arytmetyczno-logiczny posiadający w danej chwili najwyższy priorytet i będący w stanie przetwarzania tzn. taki, który nie jest zablokowany ani zatrzymany. Procesor wirtualny wykorzystujący aktualnie procesor fizyczny zajmuje go co najmniej na czas wykonywania (interpretowania) jednego rozkazu WIK-a.

#### 2.4. Struktura i działanie wirtualnego procesora arytmetyczno-logicznego; lista rozkazów

Wirtualny procesor arytmetyczno-logiczny służy do wykonywania (interpretowania) rozkazów WIK-a. Podstawowymi rejestrami wirtualnego procesora arytmetyczno-logicznego są:

- 16-bitowy rejestr A wykorzystywany jako akumulator w operacjach arytmetyczno-logicznych,
- 16-bitowy rejestr A' wykorzystywany jako "przedłużenie" rejestru A w niektórych operacjach arytmetycznych,
- 16-bitowy rejestr X wykorzystywany do X-modyfikacji wykonywanej w odniesieniu do rozkazów adresowych,
- 10-bitowy rejestr LR wykorzystywany jako licznik rozkazów i wskazujący adres względny następnego rozkazu do wykonania,

- cztery 16-bitowe rejestry bazowe (S-rejestry) wykorzystywane do adresacji pamięci wirtualnej opisanej w punkcie 2.2;
- 8-bitowy rejestr stanu procesora SW.

Poszczególne bity rejestru SW oznaczają:

- bit nr 0 - wskaźnik przeskoku P; gdy zawartość P równa się jeden, to zawartość rejestru LR zwiększona jest o długość następnego rozkazu (liczoną w słowach), a zawartość P zmieniana jest na zero,
- bit nr 1 - wskaźnik przeniesienia C ustawiany zgodnie z przeniesieniem z zerowej pozycji akumulatora w czasie wykonywania niektórych rozkazów arytmetyczno-logicznych,
- bity nr 2,3,4 - wskaźniki L,E,G ustawiane rozkazem porównania arytmetycznego (CO),
- bit nr 5 - wskaźnik nadmiaru V ustawiany na wartość równą jeden w przypadku przekroczenia dozwolonego zakresu liczb w czasie wykonywania niektórych rozkazów arytmetycznych,
- bit nr 6 - wskaźnik zatrzymania Z; gdy zawartość Z jest równa jeden, procesor jest w stanie zatrzymania,
- bit nr 7 - wskaźnik zablokowania B; gdy zawartość B jest równa jeden, procesor jest w stanie zablokowania.

Ze względu na sposób wykorzystania rozkazy wykonywane przez wirtualny procesor arytmetyczno-logiczny dzielą się na:

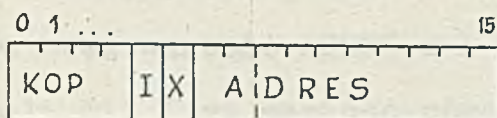
- rozkazy arytmetyczno-logiczne,
- rozkazy synchronizacyjne.

Rozkazy arytmetyczno-logiczne dzielą się na:

- rozkazy adresowe,
- rozkazy z operandem bezpośrednim,
- rozkazy z krótkim argumentem,
- rozkazy bezargumentowe krótkie,
- rozkazy bezargumentowe długie.

## Rozkazy adresowe

Format rozkazów adresowych przedstawiono na rysunku 2.



Rys.2. Format rozkazów adresowych

KOP - kod operacji przyjmujący wartości od 2 do 15;

KOP=0 wykorzystywany jest do kodowania rozkazów synchronizacyjnych;

KOP=1 - do kodowania pozostałych grup rozkazów arytmetyczno-logicznych;

I - bit adresacji pośredniej;

X - bit X-modyfikacji;

ADRES - 10-bitowy adres względny.

Wykonanie rozkazu adresowego składa się z następujących etapów:

- pobrania rozkazu do niedostępnego dla programisty rejestru rozkazu RR,
- obliczenia adresu efektywnego,
- wykonania operacji określonej przez KOP,
- zwiększenia zawartości licznika rozkazów LR o dwa.

Obliczenie adresu efektywnego polega na wykonaniu wszystkich wymaganych X-modyfikacji oraz adresacji pośrednich w sposób opisany poniżej i może być czynnością wieloetapową. Wykonanie na pewnym etapie X-modyfikacji lub adresacji pośredniej zależy od wartości (na tym etapie) bitów X oraz I.

Przyjmuje się jednak, że jeżeli należy wykonać zarówno X-modyfikację jak i adresację pośrednią, to najpierw wykonuje się X-modyfikację.

X-modyfikacja jest wykonywana, jeżeli bit X ma wartość równą jeden. Polega ona na dodaniu zawartości rejestru X do wartości określonej przez ADRES. Tak otrzymana wartość traktowana jest jako nowa wartość ADRES-u i musi zawierać się w 10-ciu bitach. Jeżeli bit I ma wartość równą jeden, to wykonywana jest adresacja pośrednia. Wówczas dotychczas wyliczona wartość ADRES-u traktowana jest jako adres względny pierwszego z dwu kolejnych słów w pamięci wirtualnej wykorzystywanych do adresacji pośredniej. Cztery najbardziej prawe bity pierwszego słowa oraz wszystkie bity drugiego słowa umieszczane są w rejestrze RR i traktowane są jako nowe wartości bitów I, X oraz ADRES-u ; stosuje się do nich ponownie reguły opisane powyżej. Należy podkreślić, że wyliczanie adresu poprzez wykonywanie X-modyfikacji oraz adresacji pośredniej trwa tak długo, aż bity I oraz X nie będą miały wartości równych zero.

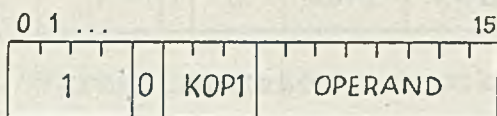
Uzyskaną w ten sposób wartość ADRES-u będziemy nazywać adresem efektywnym i będziemy oznaczać przez V. Zawartość słowa o adresie V będziemy oznaczać przez /V/.

Przez /A/, /X/, /LR/ oznaczać będziemy odpowiednio: zawartość rejestru A, zawartość rejestru X oraz zawartość rejestru LR.

Przez umieszczenie przy symbolu rejestru indeksów oznaczeń będziemy numery bitów rejestru wykorzystywanych w operacji. Poniżej podano listę rozkazów adresowych.

KOP	Nazwa rozkazu	Instyrukcja	Opis operacji
2	LA	Ładuj A	$/V/ \rightarrow A$
3	LX	Ładuj X	$/V/ \rightarrow X$
4	SA	Pamiętaj A	$/A/ \rightarrow V$
5	SX	Pamiętaj X	$/X/ \rightarrow V$
6	AD	Dodaj do A	$/A/+/V/ \rightarrow A$
7	SV	Odejmij od A	$/A/-/V/ \rightarrow A$
8			
9	ML	Mnóż logicznie przez A	$/A/\wedge /V/ \rightarrow A$
10	JP	Skocz	$V \rightarrow LR$
11	JS	Skocz ze śladem	$/LR_{0-1}/ \rightarrow V; /LR_{2-9}/ \rightarrow V+1;$ $V+2 \rightarrow LR$
12	CO	Porównaj arytmetycznie	Jeżeli $/A/ < /V/$ , to $1 \rightarrow L, 0 \rightarrow E, 0 \rightarrow G$ . Jeżeli $/A/ = /V/$ , to $0 \rightarrow L, 1 \rightarrow E, 0 \rightarrow G$ . Jeżeli $/A/ > /V/$ , to $0 \rightarrow L, 0 \rightarrow E, 1 \rightarrow G$ .
13		Do późniejszego wykorzystania	
14			
15			

Rozkazy z operandem bezpośrednim służą do umieszczania stałych w rejestrach procesora oraz do wykonywania operacji arytmetycznych, których pierwszym operandem jest zawartość jednego z rejestrów a drugi jest wskazany bezpośrednio w rozkazie. Format tych rozkazów pokazano na rysunku 3.



Rys.3. Format rozkazów z operandem bezpośrednim.

KOP1 - przedłużenie kodu operacji

OPERAND - liczba całkowita bez znaku.

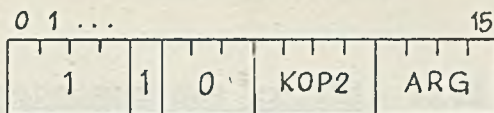
Poniżej podano listę rozkazów z operandem bezpośrednim.

KOP	Nazwa rozkazu	Instrukcja	Opis operacji
0	LAS	Ładuj krótko A	OPERAND $\rightarrow$ A <sub>8-15</sub>
1	LXS	Ładuj krótko X	OPERAND $\rightarrow$ X <sub>8-15</sub>
2	ADS	Dodaj krótko do A	/A/+ OPERAND $\rightarrow$ A
3	XDS	Dodaj krótko do X	/X/+ OPERAND $\rightarrow$ X
4	SVS	Odejmij krótko od A	/A/- OPERAND $\rightarrow$ A
5	XSS	Odejmij krótko od X	/X/- OPERAND $\rightarrow$ X
6	JPS	Skocz krótko	OPERAND $\rightarrow$ LR <sub>2-9</sub> LR <sub>0-1</sub> bez zmian
7		Do późniejszego wykorzystania	

Rozkazy z krótkim argumentem służą do logicznego, arytmetycznego i cyklicznego przesuwania zawartości rejestru A w lewo lub prawo o ilość pozycji bezpośrednio określoną w rozkazie oraz do testowania stanu klawiatury.

Format tych rozkazów pokazano na rysunku 4.





Rys.4. Format rozkazów z krótkim argumentem .

KOP2 - przedłużenie kodu operacji przyjmujące wartości od 1 do 15;

KOP2=0 wykorzystano do kodowania rozkazów bezargumentowych długich;

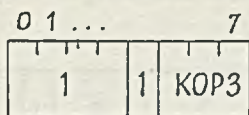
ARG - ilość pozycji przy przesunięciach lub numer przycisku klawiatury.

Listę rozkazów z krótkim argumentem przedstawiono poniżej.

KOP2	Nazwa rozkazu	Instrukcja	Opis operacji
1	LL	Przesuń A logicznie w lewo	Zawartość A zostaje przesunięta o ARG pozycji w lewo
2	LR	Przesuń A logicznie w prawo	Zawartość A zostaje przesunięta o ARG pozycji w prawo
3	CL	Przesuń A cyklicznie w lewo	Zawartość A zostaje przesunięta cyklicznie w lewo o ARG pozycji
4	CR	Przesuń A cyklicznie w prawo	Zawartość A zostaje przesunięta cyklicznie w prawo o ARG pozycji
5	AL	Przesuń A arytmetycznie w lewo	$A_{\emptyset}$ -bez zmian; $A_{1-15}$ zostaje przesunięta o ARG pozycji w lewo
6	AR	Przesuń A arytmetycznie w prawo	$A_{\emptyset}$ -bez zmian; $A_{1-15}$ zostaje przesunięta o ARG pozycji w prawo
7	K1	Przeskocz, gdy przycisk klawiatury wciśnięty	Gdy przycisk nr ARG wciśnięty, to $1 \rightarrow P$
8	K0	Przeskocz, gdy przycisk klawiatury wciśnięty	Gdy przycisk nr ARG wciśnięty, to $1 \rightarrow P$
9			
10			
11		Do późniejszego	
12		wykorzystania	
13			
14			
15			

Rozkazy bezargumentowe krótkie służą do działania na rejestrach i wskaźnikach procesora.

Format tych rozkazów pokazano na rysunku 5.



Rys.5. Format rozkazów bezargumentowych krótkich.

KOP3 - przedłużenie kodu operacji przyjmujące wartości od 1 do 7;

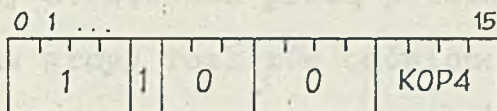
KOP3=0 wykorzystano do kodowania rozkazów z krótkim argumentem oraz bezargumentowych długich.

Listę rozkazów bezargumentowych krótkich przedstawiono poniżej.

KOP3	Nazwa rozkazu	Instrukcja	Opis operacji
1	JL	Przeskocz, gdy L=1	Jeżeli L=1, to $1 \rightarrow P$ .
2	JE	Przeskocz, gdy E=1	Jeżeli E=1 to $1 \rightarrow P$ .
3	JG	Przeskocz, gdy G=1	Jeżeli G=1, to $1 \rightarrow P$ .
4	JV	Przeskocz, gdy V=1	Jeżeli V=1, to $1 \rightarrow P$ .
5	JI	Przeskocz bezwarunkowo	$1 \rightarrow P$
6	ZA	Zeruj A	$0 \rightarrow A$
7	NA	Neguj logicznie A	$\overline{A} \rightarrow A$

Rozkazy bezargumentowe długie służą do wykonywania operacji na rejestrach procesora.

Format tych rozkazów przedstawiono na rysunku 6.



Rys.6. Format rozkazów bezargumentowych długich.

KOP4 - przedłużenie kodu operacji.

Listę rozkazów bezargumentowych długich przedstawiono poniżej.

KOP4	Nazwa rozkazu	Instrukcja	Opis operacji
0		Do późniejszego wykorzystania.	
1			
2			
3			
4	AX	Prześlij A do X	$/A/ \rightarrow X$
5	ALR	Prześlij A do LR	$/A/ \rightarrow LR$
6	CHA	Zamień słowa w A	$/A_{0-7}/ \rightarrow A_{8-15}; /A_{8-15}/ \rightarrow A_{0-7}$
7	CHX	Zamień słowa w X	$/X_{0-7}/ \rightarrow X_{8-15}; /X_{8-15}/ \rightarrow X_{0-7}$
8		Do późniejszego wykorzystania	
9			
10			
11			
12	XA	Prześlij X do A	$/X/ \rightarrow A$
13	LRA	Prześlij LR do A	$/LR/ \rightarrow A$
14			
15		Do późniejszego wykorzystania	

Rozkazy synchronizacyjne dzielą się na:

- rozkazy przesyłania informacji,
- rozkazy tworzenia procesorów,
- rozkazy sterowania pracą procesorów.

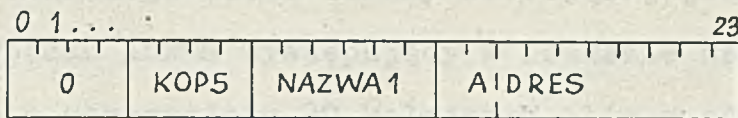
Poszczególne grupy rozkazów omówiono poniżej.

Rozkazy przesyłania informacji

Do grupy tej należą rozkazy:

WYŚLIJ KOMUNIKAT,  
CZEKAJ NA KOMUNIKAT,  
WYŚLIJ ODPOWIEDŹ,  
CZEKAJ NA ODPOWIEDŹ.

Format tych rozkazów pokazano na rysunku 7.



Rys.7. Format rozkazów przesyłania informacji.

KOP5 - przedłużenie kodu operacji obejmujące wartości od 0 do 3,

NAZWA1 - nazwa procesora lub zero,

ADRES - 10-bitowy adres względny.

Obecnie omówione zostaną poszczególne rozkazy przesyłania informacji.

WYŚLIJ KOMUNIKAT - KOP5=0, nazwa rozkazu -SEM.

Przy pomocy tego rozkazu można wysłać komunikat do dowolnego wirtualnego procesora w komputerze WIK. W tym celu procesor-nadawca komunikatu zgłasza zapotrzebowanie na wolną stronę sterującą. W przypadku braku takiej strony procesor zostaje zablokowany i umieszczony w kolejce procesorów czekających na wolne strony sterujące.

Po otrzymaniu wolnej strony sterującej umieszcza się w niej nazwę procesora-nadawcy oraz treść komunikatu.

Nazwa nadawcy komunikatu zostaje umieszczona w drugim słowie strony sterującej. Natomiast treść komunikatu może zostać umieszczona w słowach od trzeciego do trzydziestego drugiego. Zawartość trzeciego słowa strony sterującej (pierwszego słowa treści komunikatu) jest ustalana przez komputer WIK. W przypadku kiedy nadawcą komunikatu jest wirtualny procesor zewnętrzny informacje dotyczące zawartości trzeciego i dalszych słów strony sterującej podano w punkcie 2.6. Jeżeli nadawcą komunikatu jest wirtualny procesor arytmetyczno-logiczny, to zawartość trzeciego słowa strony sterującej jest równa zero. W tym przypadku ADRES występujący w rozkazie wskazuje adres pierwszego z maksymalnie 29 kolejnych słów w pamięci wirtualnej, których zawartości zostaną przepisane do strony sterującej poczynając od jej czwartego słowa. Operacja przepisywania zostaje zawsze przerwana po przepisaniu do strony sterującej słowa o zawartości równej  $12_8$  (w kodzie ISO-7 jest to znak "nowa linia").

Jeżeli spośród 29 słów, których zawartości przepisano do strony zerowej, żadne nie zawierało  $12_8$ , to zamiast zawartości słowa 29-ego na stronie sterującej zostanie zapamiętana wartość  $12_8$ . Tak przygotowaną stronę sterującą umieszcza się w kolejce komunikatów i odpowiedzi procesora-odbiorcy komunikatu. Nazwę tego procesora wskazuje w rozkazie NAZWA1. Kolejka komunikatów i odpowiedzi tworzona jest na zasadzie:

"pierwszy przyszedł - pierwszy zostanie umieszczony"

Jeżeli procesor-odbiorca komunikatu jest w stanie zablokowania z powodu oczekiwania na komunikat pochodzący właśnie od proce-

sora-nadawcy lub z powodu oczekiwania na dowolny komunikat, to zostaje on odblokowany.

Jeżeli odbiorca komunikatu nie istnieje w komputerze WIK, to procesor-nadawca otrzymuje odpowiedź generowaną przez WIK-a. Wówczas NAZWA1 określa nazwę "nadawcy" takiej odpowiedzi, ~~a treść odpowiedzi~~, a treść odpowiedzi zawarta jest w dwu słowach.

Zawartość pierwszego słowa treści odpowiedzi jest równa  $14_8$  a zawartość drugiego oznacza koniec odpowiedzi i jest równa  $12_8$ .

CZEKAJ NA KOMUNIKAT -KOP5=1, nazwa rozkazu -WME.

W pierwszym etapie wykonywania tego rozkazu sprawdza się, czy w kolejce komunikatów i odpowiedzi znajduje się komunikat, którego nadawcą jest procesor o nazwie określonej przez NAZWA1 (o ile NAZWA1 ma wartość różną od zera). Jeżeli takiego komunikatu brak, procesor zostaje zablokowany.

Jeżeli NAZWA1 ma wartość równą zero, to komunikaty w kolejce komunikatów i odpowiedzi są wykorzystywane na zasadzie "pierwszy przyszedł - pierwszy zostanie wykorzystany" a procesor zostaje zablokowany wtedy, gdy w kolejce nie ma żadnego komunikatu.

Jeżeli procesor nie zostanie zablokowany, to ADRES występujący w rozkazie wskazuje adres pierwszego z maksymalnie 30 kolejnych słów w pamięci wirtualnej, do których zostanie przepisana treść komunikatu. Operacja przepisania zostaje przerwana po przepisaniu słowa o zawartości równej  $12_8$ -oznaczającej koniec komunikatu. W ten sposób treść komunikatu (zawarta w maksymalnie 30 słowach) jest udostępniona programiście.

Należy podkreślić, że po wykonaniu tego rozkazu strona sterująca, którą wykorzystano do przesłania komunikatu nie staje się wolną

stroną sterującą, pozostaje nadal w kolejce komunikatów i odpowiedzi, a jej zawartość traktowana jest nadal jako komunikat.

WYŚLIJ ODPOWIEDŹ - KOP5=2, nazwa rozkazu -SAN.

Rozkaz ten służy do wysyłania odpowiedzi do wirtualnych procesorów w komputerze WIK.

Procesor może wysyłać odpowiedzi tylko do takich wirtualnych procesorów, od których otrzymał komunikaty. Ilość wysłanych odpowiedzi do danego wirtualnego procesora nie może być większa od ilości otrzymanych od niego komunikatów. Do wysyłania kolejnych odpowiedzi do danego wirtualnego procesora używa się stron sterujących, przy pomocy których otrzymano od niego kolejne komunikaty. Nazwę procesora-nadawcy odpowiedzi umieszcza się w drugim słowie strony sterującej. W słowach od trzeciego do trzydziestego drugiego umieszcza się treść odpowiedzi. Zawartość pierwszego słowa treści odpowiedzi (trzeciego słowa strony sterującej) jest ustalana przez WIK-a. W przypadku, gdy nadawcą odpowiedzi jest wirtualny procesor arytmetyczno-logiczny zawartość pierwszego słowa treści odpowiedzi jest równa zero.

O zawartości pozostałych słów treści odpowiedzi decyduje programista. Wówczas ADRES występujący w rozkazie wskazuje adres pierwszego z maksymalnie 29 kolejnych słów w pamięci wirtualnej, których zawartości zostaną przepisane do strony sterującej poczynając od jej czwartego słowa. Przepisanie do strony sterującej słowa o zawartości równej  $12_8$  kończy operację przepisywania. Jeżeli wśród 29 przepisanych słów żadne nie zawierało  $12_8$ , to zamiast zawartości słowa dwudziestego dziewiątego na stronie sterującej zostanie zapamiętana wartość  $12_8$ .

Informacje dotyczące tworzenia odpowiedzi przez wirtualne procesory we/wy i wirtualne procesory zewnętrzne podano w punktach 2.5 i 2.6.

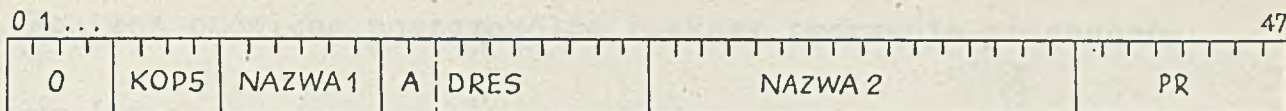
Tak przygotowaną stronę sterującą umieszcza się w kolejce komunikatów i odpowiedzi związanej z procesorem - odbiorcą odpowiedzi. Jeżeli procesor-odbiorca odpowiedzi znajduje się w stanie zablokowania z powodu oczekiwania na odpowiedź pochodzącą właśnie od procesora-nadawcy, to zostaje on odblokowany. Nazwę procesora-odbiorcy określa w rozkazie NAZWA 1 (o ile ma wartość różną od zera). Jeżeli NAZWA1 ma wartość równą zero, to odpowiedź jest wysyłana do procesora, który przysłał komunikat będący aktualnie pierwszym komunikatem w kolejce komunikatów i odpowiedzi. Jeżeli procesor-odbiorca nie istnieje w komputerze WIK, to strona sterująca przeznaczona do przesłania odpowiedzi staje się wolną stroną sterującą.

CZEKAJ NA ODPOWIEDŹ - KOP5=3 , nazwa rozkazu WAN

Pierwszym etapem wykonania tego rozkazu jest sprawdzenie, czy w kolejce komunikatów i odpowiedzi znajduje się odpowiedź, której nadawcą jest procesor o nazwie określonej przez NAZWA1. Jeżeli takiej odpowiedzi nie ma, procesor zostaje zablokowany. Przyczyną zablokowania jest oczekiwanie na odpowiedź pochodzącą od procesora o nazwie NAZWA1. Jeżeli procesor nie zostanie zablokowany, wówczas ADRES wskazuje adres pierwszego z maksymalnie 30 kolejnych słów w pamięci wirtualnej, do których zostanie przepisana treść odpowiedzi. Po przepisaniu słowa o zawartości równej 12<sub>g</sub> operacja przepisywania treści odpowiedzi zostaje zakończona. Po wykonaniu wyżej opisanych czynności strona sterująca, którą wykorzystano do przesłania odpowiedzi staje się wolną stroną sterującą.







Rys.9. Format rozkazu UTWORZ PROCESOR ARYTMETYCZNO-LOGICZNY  
(w przypadku, gdy pamięć wirtualną zrealizowano w oparciu o pamięć operacyjną i pamięci dyskowe).

KOP5 - przedłużenie kodu operacji przyjmujące wartość równą 4,

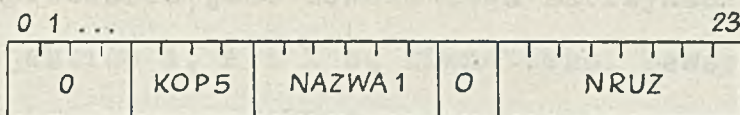
NAZWA1 - nazwa tworzonego procesora,

ADRES - 10-bitowy adres względny,

NAZWA2 - nazwa segmentu (16-bitowa) będącego początkową pamięcią względną tworzonego procesora,

PR - priorytet tworzonego procesora przyjmujący wartości od 0 do 255.

Format pozostałych rozkazów z tej grupy pokazano na rysunku 10.



Rys.10. Format rozkazów tworzenia procesorów:

wejścia, wyjścia oraz zewnętrznych.

KOP5 - przedłużenie kodu operacji przyjmujące wartości 5, 6 i 7,

NAZWA1 - nazwa tworzonego procesora,

NRUZ - numer procesora zewnętrznego związanego z tworzonym procesorem.

Poniżej omówiono poszczególne rozkazy tworzenia procesorów.

UTWÓRZ PROCESOR ARYTMETYCZNO-LOGICZNY - KOP5=4 , nazwa rozkazu CAP.

W pierwszym etapie wykonywania tego rozkazu sprawdza się, czy nazwa tworzonego procesora jest różna od nazw wszystkich procesorów wirtualnych aktualnie istniejących w komputerze WIK. Jeżeli warunek ten nie jest spełniony, to sygnalizowany jest błąd a procesor tworzący nowy wirtualny procesor zostaje zatrzymany.

W przeciwnym wypadku procesor wykonujący operację tworzenia procesora zgłasza zapotrzebowanie na wolną stronę sterującą. Jeżeli takiej strony nie ma, procesor zostaje zablokowany i umieszczony w kolejce procesorów czekających na wolne strony sterujące.

Otrzymaną wolną stronę sterującą przeznacza się do przechowywania informacji o utworzonym procesorze. Początkowym stanem utworzonego procesora jest zawsze stan zatrzymania. Początkowe zawartości rejestrów A, A' i X są równe zero. Zawartość rejestru LR jest określona przez ADRES podany w rozkazie. Dwa najbardziej lewe bity ADRES-u wskazują S-rejestr, w którym zostaje umieszczona nazwa segmentu określona przez NAZWA2. Pozostałe rejestry bazowe są nieokreślone. Wskaźniki P,C,L,E,G,V,B rejestru SW są równe zero.

UTWÓRZ PROCESOR WEJŚCIA - KOP5=5, nazwa rozkazu - CIP

W pierwszym etapie wykonywania tego rozkazu sprawdza się, czy nazwa tworzonego procesora jest różna od nazw wszystkich procesorów aktualnie istniejących w komputerze WIK. Jeżeli warunek ten nie jest spełniony, to sygnalizowany jest błąd a procesor wykonujący operację tworzenia procesora zostaje zatrzymany.

Jak wspomniano w punkcie 2.1 w komputerze WIK z każdym wirtualnym procesorem wejścia związane jest w sposób wzajemnie jednoznaczny jedno urządzenie zewnętrzne. (Uwaga ta nie dotyczy kanałów bezpośredniego dostępu).

Każdemu urządzeniu zewnętrznemu, które można dołączyć do systemu Momik 8b, w komputerze WIK został przyporządkowany 8-bitowy numer (NRUZ). Numer urządzenia zewnętrznego (NRUZ) jest identyczny z numerem linii przerwań w systemie Momik 8b, do której urządzenie zewnętrzne zostało dołączone. Podczas wykonywania operacji tworzenia procesora sprawdza się, czy zasada wzajemnej jednoznaczności jest przestrzegana. Jeżeli tak nie jest, to sygnalizowany jest błąd, operacja tworzenia procesora zostaje przerwana a procesor wykonujący tę operację zostaje zatrzymany. Jeżeli obydwa wyżej wymienione warunki są spełnione, to procesor wykonujący operację tworzenia procesora zgłasza zapotrzebowanie na wolną stronę sterującą. Jeżeli takiej strony nie ma, procesor zostaje zablokowany i umieszczony w kolejce procesorów czekających na wolne strony sterujące. Otrzymałą stronę sterującą przeznacza się do przechowywania informacji o utworzonym procesorze. Początkowym stanem utworzonego procesora jest stan zatrzymania; rejestry bazowe są nieokreślone a zawartości pozostałych rejestrów są równe zero.

UTWÓRZ PROCESOR WYJŚCIA - KOP5=6, nazwa rozkazu - COP

Rozkaz ten wykonany jest w taki sam sposób jak rozkaz UTWÓRZ PROCESOR WEJŚCIA.

UTWÓRZ PROCESOR ZEWNĘTRZNY - KOP5=7, nazwa rozkazu - CEP.

Pozkaz ten wykonywany jest w sposób analogiczny jak rozkaz UTWÓRZ PROCESOR WEJŚCIA.

Rozkazy sterowania pracą procesorów

Do tej grupy należą następujące rozkazy:

USUŃ PROCESOR,

STARTUJ PROCESOR,

STARTUJ PROCESOR OD PODANEGO ADRESU,

ZATRZYMAJ PROCESOR,

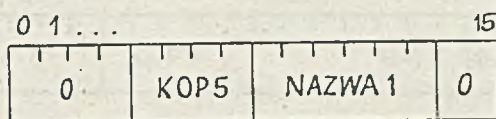
ŁADUJ S-REJESTR NR 0,

ŁADUJ S-REJESTR NR 1,

ŁADUJ S-REJESTR NR 2,

ŁADUJ S-REJESTR NR 3.

Rozkazy USUŃ PROCESOR, STARTUJ PROCESOR oraz ZATRZYMAJ PROCESOR mają format pokazany na rysunku 11.

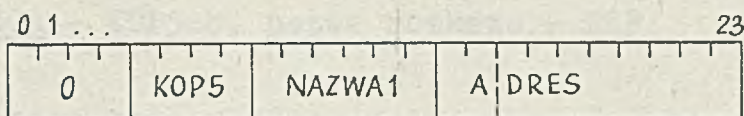


Rys.11. Format rozkazów USUŃ PROCESOR, STARTUJ PROCESOR, ZATRZYMAJ PROCESOR.

KOP5 - przedłużenie kodu operacji przyjmujące wartości 8,9,11

NAZWA1 - nazwa procesora.

Format rozkazu STARTUJ PROCESOR OD PODANEGO ADRESU pokazano na rysunku 12.



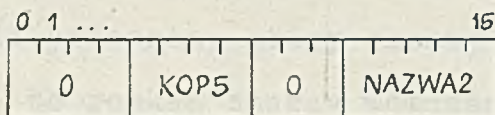
Rys.12. Format rozkazu STARTUJ PROCESOR OD PODANEGO ADRESU.

KOP5 - przedłużenie kodu operacji przyjmujące wartości równą 10,

NAZWA 1 - nazwa procesora,

ADRES - 10-bitowy adres względny.

W przypadku zrealizowania pamięci wirtualnej w oparciu tylko o pamięć operacyjną systemu Momik 8b rozkazy ładowania S-rejestrów mają format pokazany na rysunku 13.

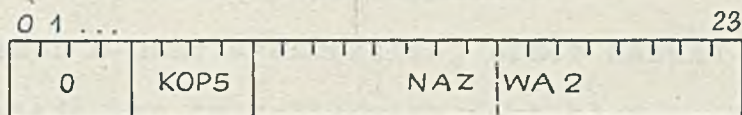


Rys.13. Format rozkazów ładowania S-rejestrów.

KOP5 - przedłużenie kodu operacji przyjmujące wartości 12,13,14,15,

NAZWA2 - nazwa segmentu .

Jeżeli pamięć wirtualna została zrealizowana w oparciu o pamięć operacyjną i pamięci dyskowe, to rozkazy ładowania S-rejestrów mają format pokazany na rysunku 14.



Rys.14. Format rozkazów ładowania S-rejestrów w przypadku zrealizowania pamięci wirtualnej w oparciu o pamięć operacyjną i pamięci dyskowe .

KOP5 - przedłużenie kodu operacji przyjmujące wartości 12,13,14,15,

NAZWA2 - 16-bitowa nazwa segmentu.

Poniżej omówiono poszczególne rozkazy sterowania pracą procesorów.

USUŃ PROCESOR - KOP5=8, nazwa rozkazu - DEP.

Przy pomocy tego rozkazu można usunąć z komputera WIK wirtualny procesor, który znajduje się w stanie zatrzymania lub zablokowania i zatrzymania. Jeżeli procesor w żadnym z tych stanów się nie znajduje, sygnalizowany jest błąd, a procesor wykonujący ten rozkaz zostaje zatrzymany.

Po usunięciu procesora z komputera WIK strona sterująca, która służyła do przechowywania o nim informacji staje się wolną stroną sterującą. Nazwę procesora, który należy usunąć określa NAZWA1. Jeżeli procesor, który chcemy usunąć w komputerze WIK nie istnieje, to rozkaz znaczy wówczas "nic nie rób".

STARTUJ PROCESOR - KOP5=9, nazwa rozkazu - ST1.

Po wykonaniu tego rozkazu procesor o nazwie określonej przez NAZWA1 będzie w stanie:

- przetwarzania - jeśli był w stanie przetwarzania lub zatrzymania,
- zablokowania - jeśli był w stanie zablokowania lub zablokowania i zatrzymania.

Jeżeli procesor o nazwie określonej przez NAZWA1 w komputerze WIK nie istnieje, to sygnalizowany jest błąd a procesor wykonujący ten rozkaz zostaje zatrzymany.

STARTUJ PROCESOR OD PODANEGO ADRESU - KOP5=10, nazwa rozkazu - ST2.

Wykonanie tego rozkazu różni się od wykonania rozkazu STARTUJ PROCESOR w sposób następujący:

- rozkaz ten można stosować tylko w odniesieniu do wirtualnych procesorów arytmetyczno-logicznych,
- procesor o nazwie określonej przez NAZWA1 musi być w stanie zatrzymania lub w stanie zablokowania i zatrzymania,
- po wykonaniu tego rozkazu licznik rozkazów (LR) procesora o nazwie określonej przez NAZWA1 przyjmuje wartość określoną przez ADRES.

ZATRZYMAJ PROCESOR - KOP5=11, nazwa rozkazu - STP.

Po wykonaniu tego rozkazu procesor o nazwie określonej przez NAZWA1 będzie w stanie:

zatrzymania - jeśli był w stanie przetwarzania lub zatrzymania,

zablokowania i zatrzymania - jeśli był w stanie zablokowania lub zablokowania i zatrzymania.

Jeżeli procesor o nazwie określonej przez NAZWA1 w komputerze WIK nie istnieje, to sygnalizowany jest błąd, a procesor wykonujący ten rozkaz zostaje zatrzymany.

ŁADUJ S-REJESTR NR 0 - KOP5=12, nazwa rozkazu - LS0.

Przy pomocy tego rozkazu w rejestrze bazowym nr 0, procesora aktualnie wykonującego ten rozkaz, umieszcza się nazwę segmentu określoną przez NAZWA2. Jeżeli S-rejestr nr 0 był nieokreślony, to po wykonaniu tego rozkazu staje się on określony i może być wykorzystywany przy adresacji pamięci wirtualnej.

ŁADUJ S-REJESTR NR 1 - KOP5=13, nazwa rozkazu - LS1.

Rozkaz ten wykonywany jest w sposób analogiczny jak rozkaz ŁADUJ S-REJESTR NR 0.

ŁADUJ S-REJESTR NR 2 - KOP5=14, nazwa rozkazu - LS2.

Rozkaz ten wykonywany jest w sposób analogiczny jak rozkaz ŁADUJ S-REJESTR NR 0.

ŁADUJ S-REJESTR NR 3 - KOP5=15, nazwa rozkazu - LS3.

Rozkaz ten wykonywany jest w sposób analogiczny jak rozkaz ŁADUJ S-REJESTR NR 0.



## 2.5. Struktura i działanie wirtualnego procesora we/wy

Wirtualny procesor we/wy (wejścia/wyjścia) służy do prowadzenia transmisji danych pomiędzy urządzeniem zewnętrznym z nim związanym a pamięcią wirtualną oraz może być wykorzystywany jako wirtualny zegar czasu rzeczywistego.

Jako pierwszy zostanie omówiony wirtualny procesor we/wy służący do prowadzenia transmisji danych.

Transmisje danych prowadzone przez wirtualny procesor we/wy dzielą się na dwa rodzaje:

- transmisje tekstowe,
- transmisje binarne.

Transmisja typu tekstowego polega na tym, że do pamięci wirtualnej przesyłane są tylko te znaki, których kody są różne od 0 i  $377_8$ .

W przypadku transmisji typu binarnego do pamięci wirtualnej przesyłane są wszystkie znaki począwszy od pewnego wyróżnionego znaku. Transmisje danych prowadzone są przez wirtualny procesor we/wy w oparciu o komunikaty znajdujące się w kolejce komunikatów i odpowiedzi z nim związanej. Każdy z komunikatów zawiera wówczas informacje o rodzaju transmisji, ilości znaków, które należy przetransmitować oraz adresach słów w pamięci wirtualnej, które biorą udział w transmisji. W oparciu o jeden komunikat może być przeprowadzona tylko jedna transmisja danych. Pierwszym etapem realizacji transmisji danych jest zawsze interpretacja treści komunikatu, w oparciu o który transmisja ma być przeprowadzona. Podstawowymi rejestrami wirtualnego procesora we/wy służącego do prowadzenia transmisji danych są:

- 16-bitowy licznik transmisji LT,
- 10-bitowy rejestr adresu transmisji AT,

- cztery 16-bitowe rejestry bazowe (S-rejestry),
- 8-bitowy rejestr ZT określający rodzaj transmisji,
- 8-bitowy rejestr stanu procesora SW.

Licznik transmisji LT określa ilość znaków, które pozostały do przetransmitowania pomiędzy pamięcią wirtualną a urządzeniem zewnętrznym. Przed rozpoczęciem transmisji zawartość LT ustalana jest na podstawie drugiego i trzeciego słowa treści komunikatu aktualnie interpretowanego.

10-bitowy rejestr AT określa kolejne słowo w pamięci wirtualnej biorące udział w transmisji. Zawartość rejestru AT ustalana jest na podstawie czwartego i piątego słowa treści komunikatu aktualnie interpretowanego. Po zakończeniu transmisji kolejnego znaku do (z) pamięci wirtualnej zawartość rejestru AT jest zwiększana o jeden.

Obliczanie adresu słowa w pamięci wirtualnej, biorącego udział w transmisji, odbywa się z wykorzystaniem rejestrów bazowych wirtualnego procesora we/wy w sposób opisany w punkcie 2.2.

Zawartość S-rejestrów wirtualnego procesora we/wy w czasie trwania transmisji jest identyczna z zawartością S-rejestrów nadawcy komunikatu aktualnie interpretowanego. Identyczność ta dotyczy chwili wysłania komunikatu.

8-bitowy rejestr ZT jest wykorzystywany w czasie transmisji tylko wtedy gdy zawartość szóstego słowa treści komunikatu aktualnie interpretowanego jest różna od  $12_8$ .

Wówczas zawartość rejestru ZT w chwili inicjacji transmisji jest identyczna z zawartością tego słowa. W przeciwnym wypadku jest nieokreślona.

8-bitowy rejestr SW określa stan procesora. W szczególności bity o numerach 6 i 7 sygnalizują stan zatrzymania i blokady procesora.

Zatrzymanie procesora we/wy nie oznacza przerwania aktualnie prowadzonej transmisji a jedynie zakaz rozpoczynania nowych czynności po jej zakończeniu. Czynności te scharakteryzowano w opisie cyklu pracy procesora.

Przyczyną zablokowania wirtualnego procesora we/wy może być tylko oczekiwanie na komunikat. Cykl pracy wirtualnego procesora we/wy można przedstawić następująco:

A : CZEKAJ NA KOMUNIKAT (dowolny) ;  
REALIZUJ TRANSMISJĘ WE/WY ;  
WYŚLIJ ODPOWIEDŹ ;  
SKOCZ DO A .

Wszystkie instrukcje z wyjątkiem "REALIZUJ TRANSMISJĘ WE/WY" mają takie znaczenie jak odpowiednie rozkazy WIK-a.

Instrukcja REALIZUJ TRANSMISJĘ WE/WY wykonywana jest w dwu etapach.

Pierwszym etapem realizacji transmisji we/wy jest interpretacja komunikatu, w oparciu o który transmisja zostanie przeprowadzona. Treść komunikatu wykorzystywana jest, w sposób już opisany, przy ustalaniu zawartości rejestrów.

Przebieg drugiego etapu tzn. właściwa transmisja danych zależy od:

- sprawności urządzenia zewnętrznego wykorzystywanego do transmisji,
- zawartości rejestru ZT,
- kierunku transmisji.

Jeżeli w czasie trwania transmisji wykryta zostanie niesprawność urządzenia zewnętrznego, realizacja transmisji zostaje przerwana a pierwsze słowo treści odpowiedzi zawiera informację o rodzaju niesprawności urządzenia zewnętrznego.

Związek między rodzajem niesprawności urządzenia zewnętrznego a zawartością pierwszego słowa treści odpowiedzi jest następujący:

- zawartość równa 1 oznacza nieoperatywność urządzenia zewnętrznego (brak napięć zasilających, odłączenie urządzenia od jednostki sterującej tip.),
- zawartość równa 2 oznacza błąd (przekłamanie) w trakcie wykonywania transmisji,
- zawartość równa 3 oznacza próbę realizacji transmisji wyjścia z wykorzystaniem urządzenia zewnętrznego nie wykonującego takiego rodzaju transmisji.

Jeżeli przez cały czas trwania transmisji urządzenie zewnętrzne jest sprawne, zawartość pierwszego słowa treści odpowiedzi jest równa zero. Zawartość drugiego słowa treści odpowiedzi jest zawsze równa 12<sub>8</sub>. Wpływ zawartości rejestru ZT na przebieg transmisji rozpatrzymy oddzielnie dla wirtualnego procesora wejścia i wirtualnego procesora wyjścia.

#### Wirtualny procesor wejścia

Jeżeli zawartość rejestru ZT jest nieokreślona, to znaki o kodach równych 0 i 377<sub>8</sub> są pomijane i nie są przesyłane do pamięci wirtualnej. Jeżeli zawartość rejestru ZT jest określona, to znaki o kodach równych 0 są pomijane tylko do momentu pojawienia się pierwszego znaku o kodzie różnym od zera. Jeżeli kod tego znaku jest różny od zawartości rejestru ZT, to transmisja zostaje przerwana a pierwsze słowo treści odpowiedzi zawiera wartość równą 4. W przeciwnym wypadku znak ten jest pomijany a począwszy od następnego znaku wszystkie znaki są przesyłane do pamięci wirtualnej.

### Wirtualny procesor wyjścia

Jeżeli zawartość rejestru ZT jest określona, to pierwszym znakiem wyprowadzanym na nośnik informacji jest znak o kodzie równym zawartości rejestru ZT. Jeżeli zawartość rejestru ZT jest nieokreślona, to operacja ta nie jest wykonywana a na nośnik informacji wyprowadzane są tylko zawartości słów pamięci wirtualnej.

Zatem, jeżeli zawartość rejestru ZT jest określona, to wirtualny procesor we/wy prowadzi transmisję typu binarnego. Jeżeli zawartość rejestru ZT jest nieokreślona, to wirtualny procesor we/wy prowadzi transmisję typu tekstowego.

Wirtualny procesor we/wy może być wykorzystywany jako wirtualny zegar czasu rzeczywistego tylko wtedy, gdy jest związany z urządzeniem zewnętrznym o numerze (NRUZ) równym 29.

Wówczas podstawowymi rejestrami procesora są:

- 16-bitowy rejestr czasu rzeczywistego RTR,
- 8-bitowy rejestr stanu procesora SW.

Zawartość rejestru RTR (traktowana jako liczba całkowita bez znaku) jest cyklicznie zwiększana o jeden. Częstotliwość zmian zawartości rejestru RTR jest ustalana przez operatora.

Jeżeli zawartość rejestru RTR jest równa  $2^{16}-1$  (wszystkie bity rejestru są równe 1), to kolejną zawartością rejestru jest 0.

Rejestr SW określa stan procesora. W szczególności bit nr 7 sygnalizuje stan zatrzymania procesora. Z procesorem we/wy wykorzystywanym jako wirtualny zegar czasu rzeczywistego związana jest (jak z każdym innym procesorem w komputerze WIK) kolejka komunikatów i odpowiedzi. Działanie procesora we/wy jako wirtualnego zegara czasu rzeczywistego polega na cyklicznym przeglądaniu kolejki komunikatów i odpowiedzi i porównywaniu zawartości

drugich i trzecich słów treści komunikatów z zawartością rejestru RTR.

Jeżeli zawartości drugiego i trzeciego słowa treści komunikatu są identyczne z zawartością rejestru RTR, to do procesora-nadawcy komunikatu wysyłana jest odpowiedź. Pierwsze słowo treści odpowiedzi zawiera zero, a drugie  $12_8$ . W ten sposób interpretowane są wszystkie komunikaty znajdujące się w kolejce komunikatów i odpowiedzi. Stan zatrzymania procesora oznacza, że odpowiedzi nie są wysyłane. Kolejka komunikatów i odpowiedzi przeglądana jest oddzielnie dla każdej zawartości rejestru RTR.

## 2.6. Struktura i działanie wirtualnego procesora zewnętrznego

Wirtualny procesor zewnętrzny reprezentuje w komputerze wirtualnym WIK obiekt zewnętrzny, który współpracuje z innymi procesorami wirtualnymi drogą wymiany informacji.

Z punktu widzenia programisty - użytkownika oznacza to, że:

- wysłanie informacji do obiektu zewnętrznego jest równoznaczne z wysłaniem komunikatu do reprezentującego go wirtualnego procesora zewnętrznego,
- otrzymanie komunikatu od wirtualnego procesora zewnętrznego jest równoznaczne z otrzymaniem informacji od obiektu, który jest reprezentowany przez ten procesor zewnętrzny.

Podstawową cechą rozważanego obiektu zewnętrznego jest zdolność zgłaszania (w sposób niesterowany przez WIK-a) wirtualnemu procesorowi zewnętrznemu (który reprezentuje obiekt w komputerze) chęci przesłania informacji do innego procesora wirtualnego.

Przykładami obiektów zewnętrznych mogą być: operator, inny komputer współpracujący z WIK-iem, urządzenie pomiarowe wysyłające wyniki pomiarów w sposób nie sterowany WIK-iem (np. losowy) i.t.p. Zorganizowanie wymiany informacji pomiędzy obiektem zewnętrznym a procesorami wirtualnymi - to główne zadanie wirtualnego procesora zewnętrznego. Wymiana informacji pomiędzy obiektem zewnętrznym a reprezentującym go wirtualnym procesorem zewnętrznym jest możliwa poprzez wykorzystanie urządzenia zewnętrznego spełniającego warunki dołączenia do systemu Momik 8b.

Na przykład, gdy obiektem zewnętrznym jest operator - urządzeniem zewnętrznym jest maszyna do pisania. Urządzenie zewnętrzne musi posiadać ponadto następujące własności:

- umożliwiać przesyłanie informacji w obydwu kierunkach,
- umożliwiać procesorowi zewnętrznemu dokonywanie wyboru kierunku przesyłania informacji z jednozesną blokadą kierunku przeciwnego.

Własność reagowania na blokadę kierunku transmisji powinien posiadać również obiekt zewnętrzny. W szczególnym przypadku obiekt i urządzenie zewnętrzne mogą być jednym urządzeniem fizycznym.

W aktualnie realizowanej wersji WIK-a przyjęto, że wszystkie obiekty zewnętrzne mogą wysyłać informację tylko do jednego, z góry ustalonego wirtualnego procesora arytmetyczno-logicznego o nazwie równej 1. Wirtualne procesory arytmetyczno-logiczne mogą natomiast wysyłać informację do dowolnego obiektu zewnętrznego.

Podstawowym rejestrem wirtualnego procesora zewnętrznego jest 8-bitowy rejestr stanu SW. W szczególności bity SW o numerach 4,6,7 mają następujące znaczenie:

- bit nr 4 sygnalizuje, że obiekt zewnętrzny chce przesłać informację do jednego z procesorów wirtualnych,
- bit nr 6 sygnalizuje stan zatrzymania procesora,
- bit nr 7 sygnalizuje stan zablokowania procesora.

Obiekt zewnętrzny może zgłaszać chęć przesłania informacji tylko wtedy, gdy wejściowy (tzn. od obiektu do procesora zewnętrznego) kierunek przesyłania informacji nie jest zablokowany. W tym celu powinien przesyłać do wirtualnego procesora zewnętrznego, wykorzystując urządzenie zewnętrzne z tym procesorem związane, znak o kodzie równym 52<sub>g</sub> (w kodzie ISO-7 jest to znak „\*”). Jeżeli operacja ta została wykonana poprawnie, bit nr 4 rejestru SW przyjmuje wartość równą 1, a procesor zewnętrzny zgłasza zapotrzebowanie na wolną stronę sterującą. Jeżeli wolnej strony sterującej nie ma - procesor zostanie zablokowany. W czasie trwania takiej blokady procesora zewnętrznego zablokowany jest również wejściowy kierunek transmisji.

Jeżeli w czasie trwania transmisji pomiędzy obiektem a procesorem zewnętrznym procesor zostanie wprowadzony w stan zatrzymania (rozkazem STP), to nie oznacza to przerwania transmisji a jedynie zakaz rozpoczynania nowych czynności po jej zakończeniu.

Czynności te scharakteryzowano w opisie cyklu pracy procesora.

Wirtualny procesor zewnętrzny może zostać zablokowany z następujących powodów:

- braku wolnej strony sterowania,
- braku odpowiedzi od procesora wirtualnego, do którego wysłał komunikat.

Cykl pracy wirtualnego procesora zewnętrznego można przedstawić następująco:



A : Jeżeli obiekt zewnętrzny zgłosił chęć przesłania informacji, to

REALIZUJ TRANSMISJĘ WEJŚCIA ;

WYŚLIJ KOMUNIKAT ;

CZEKAJ NA ODPOWIEDŹ ;

SKOCZ DO A .

W przeciwnym wypadku :

jeżeli dowolny procesor wirtualny przysłał komunikat, to

REALIZUJ TRANSMISJĘ WYJŚCIA ;

WYŚLIJ ODPOWIEDŹ ;

SKOCZ DO A ;

jeżeli nie nadesłano żadnego komunikatu, to

SKOCZ DO A .

Wszystkie instrukcje z wyjątkiem "REALIZUJ TRANSMISJĘ WEJŚCIA" i "REALIZUJ TRANSMISJĘ WYJŚCIA" mają takie znaczenie jak odpowiednie rozkazy WIK-a.

Obecnie omówione zostaną: "REALIZUJ TRANSMISJĘ WEJŚCIA" oraz "REALIZUJ TRANSMISJĘ WYJŚCIA".

#### Realizacja transmisji wejścia

Transmisję wejścia inicjuje wirtualny procesor zewnętrzny po uzyskaniu wolnej strony sterującej. Wysyłane przez obiekt zewnętrzny znaki traktowane są wówczas jako kolejne słowa komunikatu, który zostanie wysłany do procesora wirtualnego, z którym obiekt współpracuje. W ten sposób jest interpretowanych co najwyżej 29 znaków.

Wysłanie przez obiekt zewnętrzny znaku o kodzie równym 12<sub>g</sub> (w kodzie ISO-7 jest to znak "nowa linia") kończy transmisję

wejścia. Od tego momentu wszystkie znaki wysyłane przez obiekt do procesora zewnętrznego aż do pojawienia się znaku o kodzie równym  $52_8$  są pomijane.

Interpretacja znaku o kodzie równym  $52_8$  została opisana wyżej. Jeżeli w trakcie realizacji transmisji wejścia wśród 29 znaków nie wystąpił znak o kodzie równym  $12_8$ , to umieszcza się go w komunikacie automatycznie zamiast znaku 29-ego.

Realizacja transmisji wejścia może zostać przerwana na skutek niesprawności urządzenia zewnętrznego wykorzystywanego do transmisji. Wówczas zawartość pierwszego słowa treści komunikatu określa rodzaj niesprawności urządzenia, a ostatnie słowo zawiera znak o kodzie równym  $12_8$ . Jeżeli realizacja transmisji przebiegała poprawnie, wówczas zawartość pierwszego słowa treści komunikatu jest równa zero.

Uwaga!

Odpowiedź nadesłana przez odbiorcę komunikatu nie jest interpretowana i jest wymagana tylko ze względu na logiczną strukturę działania WIK-a.

#### Realizacja transmisji wyjścia

Celem transmisji wyjścia jest przesłanie treści komunikatu nadesłanego przez jeden z wirtualnych procesorów do obiektu zewnętrznego. Aby uniemożliwić obiektowi zewnętrznemu przesłanie w tym samym czasie informacji w kierunku przeciwnym, kierunek wejściowy zostaje zablokowany. Następnie do obiektu zewnętrznego jest przesyłana treść komunikatu poprzedzona znakiem o kodzie równym  $52_8$ .

Fakt poprzedzania treści komunikatu znakiem o kodzie równym  $52_g$  jest szczególnie istotny przy współpracy dwu wirtualnych komputerów WIK, gdy traktują się one wzajemnie jako obiekty zewnętrzne. Realizacja transmisji wyjścia może zostać przerwana na skutek niesprawności urządzenia zewnętrznego wykorzystywanego do transmisji. Wówczas pierwsze słowo odpowiedzi określa rodzaj niesprawności. Jeżeli transmisja przebiegała poprawnie, zawartość pierwszego słowa odpowiedzi jest równa zero. Zawartość drugiego słowa treści odpowiedzi jest zawsze równa  $12_g$ .

Po zakończeniu realizacji transmisji wyjścia wymiana informacji ponownie może przebiegać w obydwu kierunkach.

Zarówno w przypadku realizacji transmisji wejścia jak i realizacji transmisji wyjścia związek między zawartością pierwszego słowa treści <sup>komunikatu lub,</sup> odpowiedzi a rodzajem niesprawności urządzenia zewnętrznego jest następujący:

- zawartość równa 1 oznacza nieoperatywność urządzenia zewnętrznego (brak napięć zasilających, odłączenie urządzenia od jednostki sterującej itp.),
- zawartość równa 2 oznacza błąd (przekłamanie) w trakcie realizacji transmisji.

W przypadku realizacji transmisji wyjścia jeżeli zawartość pierwszego słowa treści odpowiedzi jest równa 3, to oznacza to, że z procesorem zewnętrznym związane jest urządzenie zewnętrzne nie prowadzące transmisji danych w obu kierunkach.

